RECONSTRUCTING SMOOTH SURFACES FROM PARTIAL, NOISY INFORMATION

Technical Note 222

July 1980

By: Harry G. Barrow, Senior Computer Scientist
Jay M. Tenenbaum, Program Manager - Computer Vision

Artificial Intelligence Center
SRI International
Menlo Park, California 94025

SRI Projects 1019, 8682, and 4683

The work reported herein was supported by the U.S. Army Research Office under Contract No. DAAG29-79-C-0216, the National Science Foundation under Grant No. MCS-7901830, and the National Aeronautics and Space Administration under Contract No. NASW-2865.

333 Ravenswood Ave. • Menlo Park, CA 94025
(415) 326-6200 • TWX: 910-373-2046 • Telex: 334-486

ABSTRACT

Interpolating smooth surfaces from boundary conditions is a ubiquitous problem in early visual processing. We describe a solution for an important special case: the interpolation of surfaces that are locally spherical or cylindrical from initial orientation values and constraints on orientation. The approach exploits an observation that components of the unit normal vary linearly on surfaces of uniform curvature, which permits implementation using local parallel processes. Experiments on spherical and cylindrical test cases have produced essentially exact reconstructions, even when boundary values were extremely sparse or only partially constrained. Results on other test cases seem in reasonable agreement with human perception.

ACKNOWLEDGMENTS

The work reported in this paper was performed under SRI's research program in computational vision, which is jointly supported by ARPA, NSF, and NASA.

I INTRODUCTION

Surface perception plays a fundamental role in early visual processing, both in humans and machines [1, 2]. An explicit representation of surface structure is directly necessary for many low-level visual functions involved in applications such as terrain modeling, navigation, and obstacle avoidance. It is also a prerequisite for general-purpose, high-performance vision systems.

Information about surfaces comes from various sources: stereopsis, motion parallax, texture gradient, shading, and contour shape, to name a few. Information may be provided in terms of absolute or relative values of orientation or range, depending upon the nature of the source. Moreover, different techniques for extracting this information are valid in different parts of the scene. For example, inferring shape from shading is difficult on a highly textured surface, or in areas of complex illumination, while stereo information is not available in textureless areas nor areas visible only from one viewpoint. Thus, in general, evidence is incomplete, may be quite sparse (as in line drawings), and subject to noise, which leads to ambiguity.

Any attempt to derive globally consistent surface descriptions from these diverse local sources must therefore address the following basic computational problems:

(1) interpolation of sparse data

(2) smoothing of noisy data

(3) deciding which techniques are applicable in which parts of the scene

(4) integration of different types of data from different sources

(5) deciding the location and physical type of boundaries

In this paper we look mainly at the first problem, which arises in virtually all theories of low-level vision [1, 2]. We principally address the problem of reconstructing a smooth surface, given a set of initial orientation values, which may be sparse or only partially constrained.

II COMPUTATIONAL PRINCIPLES

We begin with a precise definition of the reconstruction problem in terms of input and output.

The input is assumed to be in the form of sparse arrays, containing local estimates of surface range and orientation, in a viewer-centered coordinate frame. In practice, the estimates may be clustered where the information is obtainable, such as along curves corresponding to surface boundaries. In general, they are subject to error and may be only partially constrained. For example, given a three-dimensional boundary, the surface normals are only constrained to be orthogonal to the boundary elements. We also assume that the location and nature of all surface boundaries are known, since they give rise to discontinuities of range or orientation. This last condition is required in the current implementation and is intended to be relaxed at a later date to accommodate imperfect boundary detection.

The desired output is simply filled arrays of range and surface orientation representing the most likely surfaces consistent with the input data. Refinement of hypothesized surface discontinuities is also desired. These output arrays are analogous to our intrinsic images [1] or Marr's 2.5D sketch [2].

For any given set of input data, an infinitude of possible surfaces can be found to fit arbitrarily well. Which of these is best depends upon assumptions about the nature of surfaces in the world and the image formation process. Ad hoc smoothing and interpolation schemes which are not rooted in these assumptions lead to incorrect results in simple cases. For example, given a few points on the surface of a sphere, iterative local averaging [3, 4] of range values will not recover a spherical surface.

A. Assumptions about Surfaces

The principal assumption we make about physical surfaces is that range and orientation are continuous over them. We further assume that each point on the surface is essentially indistinguishable from neighboring points. Thus, in the absence of evidence to the contrary, it follows that local surface characteristics must vary as smoothly as possible and that the total variation is minimal over the surface. Range and orientation are both defined with reference to a viewer-centered coordinate system, and so they cannot directly be the criteria for evaluating the intrinsic smoothness of hypothetical surfaces. The simplest appropriate measures involve the rate of change of orientation over the surface; principal curvatures (k1, k2), Gaussian (total) curvature (k1*k2), mean curvature (k1+k2), and variations upon them all reflect this rate of change [5]. Two reasonable definitions of smoothness of a surface are uniformity of some appropriate measure of curvature [6], or minimality of integrated squared curvature [7]. Uniformity can be defined as minimal variance or minimal integrated magnitude of gradient.

The choice of a measure and how to employ it (e.g., minimize the measure or its derivative) depends, in general, upon the nature of the process that gave rise to the surface. For example, surfaces formed by elastic membranes (e.g., soap films) are constrained to minimum energy configurations characterized by minimum area and zero mean curvature [8]; surfaces formed by bending sheets of inelastic material (e.g., paper or sheet metal) are characterized by zero Gaussian curvature [9]; surfaces formed by many machining operations (e.g., planes, cylinders, and spheres) have constant principal curvatures.

We are not prepared, at this point, to maintain that any of these measures is inherently superior, particularly because of various close relationships that exist between them. We note, for example, that minimizing the integrated square of mean curvature is equivalent to minimizing the sum of integrated squares of principal curvatures and the integrated Gaussian curvature, G, as shown by:

$$\int (k1 + k2)^2 .da = \int k1^2 .da + \int k2^2 .da + 2\int k1*k2.da$$
$$= \int k1^2 .da + \int k2^2 .da + 2\int G.da \quad (1)$$

We also note that making curvature uniform by minimizing its variance of any measure over a surface is equivalent to minimizing total squared curvature, if the integral of curvature is constant. This follows from the well-known fact that for any function, f(x),

$$\text{Variance of f} = \int (f-fbar)^2 .dx$$
$$= \int f^2 .dx - [\int f.dx]^2 / DX \quad (2)$$

On any developable surface for which Gaussian curvature, G, is everywhere zero, and on a surface for which orientation is known everywhere at its boundary (e.g., the boundary is extremal), the integral of G is constant. Thus, for such surfaces, minimizing variance of G and minimizing its integrated square are equivalent.

By itself, however, uniformity of Gaussian curvature is not sufficiently constraining. Any developable surface is perfectly uniform by this criterion, so considerable ambiguity remains, as is evident in Figure 1, where all of the developable surfaces satisfy the same boundary conditions. Thus a secondary constraint, such as uniformity of mean curvature, is required to find the smoothest developable surface.

In this paper we focus on surfaces with reasonably uniform curvature--surfaces that are locally spherical or cylindrical. We shall demand exact reconstructions for spherical and cylindrical test cases and intuitively reasonable reconstructions for other smooth surfaces. In particular, given surface orientations defined around a circular outline, corresponding to the extremal boundary of a sphere, or along two parallel lines, corresponding to the extremal boundary of a right circular cylinder, we require interpolation to yield the correct spherical or cylindrical surface, with uniform (Gaussian, mean, and principal) curvature. These cases are important because they require reconstructions that are symmetric in three dimensions and independent of viewpoint. Many simple interpolation techniques fail this test, producing surfaces that are too flat or too peaked. Given good performance on the test cases, we can expect reasonable performance in general.

III A RECONSTRUCTION ALGORITHM

Although in principle correct reconstruction for our test cases can be obtained in many ways, the complexity of the interpolation process depends critically upon the representation. For example, representing surface orientation in terms of gradient space leads to difficulties because gradient varies very nonlinearly across the image of a smooth surface, becoming infinite at extremal boundaries. We shall now propose an approach that leads to elegantly simple interpolation for our test cases.

A. Coordinate Frames

Given an image plane, we shall assume a right-handed Cartesian coordinate system with x- and y- axes lying in the plane (see Figure 2). We also assume orthogonal projection in the direction of the z-axis. Each image point (x,y) has an associated range, Z(x,y); the corresponding scene point is thus specified by

$$(x, y, Z(x,y)) \quad .$$

Each image point also has an associated unit vector that specifies the local surface orientation at the corresponding scene point:

$$N(x,y) = (Nx(x,y), Ny(x,y), Nz(x,y)) .$$

Since N is normal to the surface Z,

$$\begin{aligned} Nx/Nz &= - dZ/dx \\ \text{and} \quad Ny/Nz &= - dZ/dy . \end{aligned} \tag{3}$$

(The derivatives dZ/dx and dZ/dy correspond to p and q when the surface normal is represented in gradient space form, (p,q,-1).)

Differentiating equation (3), we obtain

$$\begin{aligned} d(Nx/Nz)/dy &= - d^2 Z/dy.dx \\ \text{and} \quad d(Ny/Nz)/dx &= - d^2 Z/dx.dy . \end{aligned} \tag{4}$$

For a smooth surface, the terms on the right of (4) are equal, hence

$$d(Nx/Nz)/dy = d(Ny/Nz)/dx . \tag{5}$$

Finally, since N is a unit vector,

$$Nx^2 + Ny^2 + Nz^2 = 1 . \tag{6}$$

B. Semicircle

Let us begin by considering a two-dimensional version of surface reconstruction. In Figure 3 observe that the unit normal to a semicircular surface cross section is everywhere aligned with the radius. It therefore follows that triangles OPQ and PST are similar, and so

$$OP : OQ : QP = PS : PT : TS \quad . \qquad (7)$$

But vector OP is the radius vector (x,z) and PS is the unit normal vector (Nx,Nz). Moreover, the length OP is constant (equal to R), and the length PS is also constant (equal to unity). Hence,

$$Nx = x/R \quad \text{and} \quad Nz = z/R \quad . \qquad (8)$$

C. Sphere

Now consider a three-dimensional spherical surface, as shown in Figure 4. Again the radius and normal vectors are aligned, and so from similar figures we have

$$Nx = x/R \quad Ny = y/R \quad \text{and} \quad Nz = z/R \ . \qquad (9)$$

The point to note is that Nx and Ny are both linear functions of x and y, and that Nz can readily be derived from Nx and Ny because vector N has unit length.

D. Cylinder

The case of the right circular cylinder is only a little more complex. In Figure 5 observe a cylinder of radius R centered upon a line in the x-y plane, inclined at an angle A to the x axis. Let d be the distance of point (x,y,0) from the axis of the cylinder. Then

$$d = y.\mathrm{Cos}\ A - x.\mathrm{Sin}\ A \qquad (10)$$

$$\text{and} \quad z^2 = R^2 - d^2 \quad . \qquad (11)$$

Let Nd be the component of vector N parallel to the x-y plane; it is clearly perpendicular to the axis of the cylinder. Now, since a cross section of the cylinder is analogous to our first, two-dimensional, case,

$$Nd = d/R \quad . \tag{12}$$

Taking components of Nd parallel to the x and y axes,

$$Nx = Nd \cdot \text{Sin } A \quad \text{and} \quad Ny = -Nd \cdot \text{Cos } A \quad . \tag{13}$$

Substituting in this equation for Nd, and then for d,

$$\begin{aligned} Nx &= (y \cdot \text{Cos } A - x \cdot \text{Sin } A) \cdot \text{Sin } A/R \\ \text{and} \quad Ny &= -(y \cdot \text{Cos } A - x \cdot \text{Sin } A) \cdot \text{Cos } A/R \quad . \end{aligned} \tag{14}$$

Observe that as for the sphere, Nx and Ny are linear functions of x and y, and that Nz can be derived from Nx and Ny.

IV INTERPOLATING SPHERICAL AND CYLINDRICAL SURFACES

From the preceding section, we can see that to interpolate values for the normal vector, on spherical and cylindrical surfaces, between points where its value is known, we need only determine the linear functions that describe the components Nx and Ny. This can be done simply from known values at any three noncollinear points. The resulting functions can be used to predict precisely values of Nx and Ny, and hence Nz also, over the entire surface. The vector field produced is guaranteed to satisfy the integrability constraint of Equation 5, as may be verified by substituting for Nx, Ny, and Nz from Equations 9 or 14 (for the sphere or cylinder, respectively) and 6. Finally, the orientation field can be integrated to recover range values.

For the special test cases, because of the global nature of the linearity of Nx and Ny, it is possible to interpolate between given boundary values, treating Nx and Ny as essentially independent variables. While in general the integrability constraint should not be ignored, in practice, since complex surfaces can often be approximated locally by spheres or cylinders, this constraint is weak and its omission does not result in significant errors.

V A COMPUTATIONAL MODEL

We have implemented a model that uses parallel local operations to derive the orientation and range over a surface from boundary values. It exploits the linearity and separability results for the test cases and extends them to arbitrary smooth surfaces.

The overall system organization is a subset of the array stack architecture first proposed in [1]. It consists conceptually of two primary arrays, one for range and the other for surface normal vectors, which are in registration with each other (and with the input image). Values at each point within an array are constrained by local processes that maintain smoothness and by processes that operate between arrays to maintain the differential/integral relationship. In general, we must be able to insert initial boundary values sparsely in both range and orientation arrays and have the system relax to fill in consistent intervening values. At present we know how to handle the restricted case where only orientation is initially specified.

VI THE INTERPOLATION PROCESS

At each point in the orientation array we can imagine a process that is attempting to make the two observable components of the normal, Nx and Ny, each vary as linearly as possible. The process looks at the values of Nx (or Ny) in a small patch surrounding the point and attempts to infer the linear function, $f = ax + by + c$, that best models Nx locally. It then tries to relax the value for the point to reduce the supposed error.

There are numerous ways to implement such a process, and we shall describe some of the ones with which we have experimented. One of the simplest is to perform a local least-squares fit, deriving the three parameters a, b, and c. The function f is then used to estimate a corrected value for the central point. The least-squares fitting process is equivalent to taking weighted averages of the values in the patch, using three different sets of weights:

$$\sum_i x_i Nx_i \ , \quad \sum_i y_i Nx_i \ , \quad \sum_i Nx_i \quad . \tag{15}$$

The three parameters of f are given by three linear combinations of these three averages.

If we are careful to use a symmetric patch with its origin at the point in question, the sets of weights and the linear combinations are particularly simple--the three sums in equation (15) correspond, respectively, to

$$a*\sum_i x_i^2 \ , \quad b*\sum_i y_i^2 \ , \quad c*\sum_i 1 \quad . \tag{16}$$

Equations (15) and (16) can be readily solved for a, b, and c; but note that under the above assumptions, f(0,0)=c, so computation of a and b is unnecessary for updating the central point, unless derivatives are also of interest.

An alternative approach follows from the fact that a linear function satisfies the equation

$$\nabla^2 f = 0 \quad . \tag{17}$$

Numerical solution of this equation, subject to boundary conditions, is well known. The ∇^2 operator may be discretely approximated by the operator

$$\begin{array}{ccc} & -1 & \\ -1 & 4 & -1 \\ & -1 & \end{array} \quad .$$

Applying this operator at a point in the image leads to an equation of the form

$$4Nx_0 - Nx_1 - Nx_2 - Nx_3 - Nx_4 = 0 \quad , \qquad (18)$$

and hence, rewriting,

$$Nx_0 = (Nx_1 + Nx_2 + Nx_3 + Nx_4)/4 \quad . \qquad (19)$$

Equation (19) is used in a relaxation process that iteratively replaces the value of Nx_0 at each point by the average of its neighbors. Although the underlying theory is different from least-squares fitting, the two methods lead to essentially the same discrete numerical implementation.

The iterative local averaging approach works well in the interior regions of a surface, but difficulties arise near surface boundaries where orientation is permitted to be discontinuous. Care must be taken to ensure that the patch under consideration does not fall across the boundary, otherwise estimation of the parameters will be in error. On the other hand, it is necessary to be able to estimate values right up to the boundary, which may, for example, result from another surface occluding the one which we are attempting to reconstruct.

The least-squares method is applicable to any shape of patch, which we can simply truncate at the boundary. However, the linear combination used to compute each parameter depends upon the particular shape, so we must either precompute the coefficients for all possible patches (256 for a 3x3 area) or resort to inverting a 3x3 matrix to derive them for each particular patch. Neither of these is attractive.

The above disadvantages can be overcome by decomposing the two-dimensional fitting process into several one-dimensional fits. We do this by considering a set of line segments passing through the central point, as shown in Figure 6. Along each line we fit a function, f = ax + c, to the data values, and thus determine a corrected value for the point. The independent estimates produced from the set of line segments can then be averaged. If the line segments are each symmetric about the central point, then the corrected central value is again simply the average of the values along the line. The principal advantage of the decomposition is that we can discard line segments which overlap a boundary, and often at least one is left to provide a corrected value. We would prefer to use short symmetric line segments, since they form a compact operator, but in order to get into corners we need also to resort to one-sided segments (which effectively extrapolate the central value). We have implemented a scheme that uses the compact symmetric operator when it can, and an asymmetric operator when this is not possible (see Figure 7).

We have experimented with a rather different technique for coping with boundary discontinuities, which is of interest because it involves multiple interrelated arrays of information. For each component of the orientation vector we introduce two auxiliary arrays containing estimates of its gradient in the x and y directions. For surfaces of uniform curvature, such as the sphere and cylinder, these gradients will be constant over the surface; and for others, we assume they will be slowly varying. To reconstruct the components of the normal, we first compute its derivatives, then locally average the derivatives, and finally reintegrate them to obtain updated orientation estimates.

Derivatives at a point are estimated by considering line segments through the point parallel to the axes. We again fit a linear function--but now we record its slope, rather than its intercept, and insert it in the appropriate gradient array. In the interior of a region we may use a symmetric line segment, and near boundaries, a one-sided segment, as before. The gradient arrays are smoothed by an operator that forms a weighted average over a patch, which may easily be truncated at a boundary. (To form the average over an arbitrarily-shaped patch, it is only necessary to compute the sum of weighted values of points within the patch and the sum of the weights, and then divide the former by the latter.) A corrected orientation value can be computed from a neighboring value by adding (or subtracting) the appropriate gradient. Each neighboring point not separated by a boundary produces such an estimate, and all the estimates are averaged.

VII ESTIMATION OF SURFACE RANGE

The process of integrating orientation values to obtain estimates of range Z is very similar to that used in reintegrating orientation gradients. We again use a relaxation technique, and iteratively compute estimates for Z from neighboring values and the local surface orientation. Here we need orientation expressed as dZ/dx and dZ/dy, which are obtained from Nx and Ny by Equation 3. At least one absolute value of Z must be provided to serve as a constant of integration. Providing more than one initial Z value constrains the surface to pass through the specified points; but since the inverse path from Z to N has not yet been implemented, the resulting range surface is not guaranteed to be consistent with the orientations.

VIII EXPERIMENTAL RESULTS

An interactive system was implemented in MAINSAIL [10] to experiment with and evaluate the various interpolation algorithms discussed above. This system includes facilities for generating quadric surface test cases, selecting interpolation options, and plotting error distributions.

A. Test Cases

How well do each of the above interpolation techniques reconstruct the test surfaces? To answer this, we performed a series of experiments in which the correct values of Nx and Ny were fixed along the extremal boundaries of a sphere or cylinder, as shown in Figure 8. The surface orientations reconstructed from these boundary conditions were compared with those of ideal spherical or cylindrical surfaces generated analytically.

The first set of experiments involved a sphere of radius 7 centered in a 17 x 17 interpolation array. We deliberately used a coarse grid to test the accuracy of the reconstruction under difficult conditions. (A coarse grid also has the experimental advantage of minimizing the number of iterations needed for convergence.) Correct values for Nx and Ny were fixed at points in the array falling just inside the circular extremal boundary of the sphere. Table I summarizes the results for this test case, using various interpolation operators.

The results on the spherical test case are almost uniformly good. In all cases, except gradient smoothing, the maximum absolute error is below one percent after 100 iterations $(-1.0 < Nx, Ny < 1.0)$. On any cross section through the sphere, the maximum error occurs approximately a quarter of the way in from both boundary points, the error being zero at the boundary points and also on the symmetry axis half way between them. We conclude that 8-connected, uniformly weighted averaging and 8-way linear interpolation/extrapolation are superior in terms of speed of convergence, with the linear operator preferred because of its advantages at boundaries and corners. These conclusions generalize to all of the test cases we have studied to date. Thus, for brevity, the experimental results that follow are reported only for the 8-way linear operator.

The second set of experiments involved a cylinder of radius 6, centered in an 8 x 8 interpolation array. Again, correct values for Nx and Ny were fixed at points in the array falling just inside the parallel lines representing the extremal boundaries of the cylinder. With the cylinder oriented parallel to the X or Y axis, the maximum absolute error in Nx or Ny after 50 iterations was .018 and the RMS average error .01 . After 100 iterations, the absolute error dropped to .0004 and the RMS average to .0002. When the major axis of the cylinder was inclined 60 degrees to the X-axis, the errors look much higher: .12 absolute and .03 RMS after 50 iterations; .108 absolute and .03 RMS after 100 iterations; .09 absolute and .02 RMS after 300 iterations. However, the errorful orientations were concentrated solely in the upper right and lower left corners of the array, where the cylinder boundary is effectively occluded by the array edge. Extrapolation of values from the central region, where the orientations are very accurate, into these partially occluded corners accounts for the slow rate of convergence. After 1,000 iterations, however, orientations are highly accurate throughout the array.

B. Other Smooth Surfaces

Given that orientations for uniformly curved surfaces can be accurately reconstructed, the obvious next question is how well the algorithms perform on other surfaces for which curvature is not globally uniform. A simple case to consider is that of an elliptical boundary. However, we immediately run into the problem of what is to be taken as the "correct" reconstruction. When people are asked what solid surface they perceive, they usually report either an elongated object or a squat object, roughly corresponding to a solid of revolution about the major or minor axis, respectively. The elongated object is preferred, and one can argue that it is more plausible on the grounds of general viewpoint (a fat, squat object looks elongated only from a narrow range of viewpoints). When presented with initial orientations for an elliptical extremal boundary (Figure 9), our algorithms reconstruct an elongated object, with approximately uniform curvature about the major axis. They, in effect, reconstruct a generalized cylinder [11], but without explicitly invoking processes to find the axis of symmetry or matching the opposite boundaries.

In a representative experiment, initial values for Nx and Ny were fixed inside an elliptic-shaped extremal boundary (major axis 15, minor axis 5). The reconstructed orientations were then compared with the orientations of the solid of revolution generated when the ellipse is rotated about its major axis. The resulting errors after 50 iterations were: for Nx, .02 maximum absolute error and .006 average RMS error; and for Ny, .005 maximum absolute and .002 RMS.

C. Occluding Boundaries

We also wish to know how well the reconstruction process performs when the orientation is not known at all boundary points. In particular, when the surface of interest is occluded by another object, the occluding boundary provides no constraints. In such cases, the orientation at the boundary must be inferred from that of neighboring points, just like at any other interior points of the surface. The 8-way linear operator will correctly handle these situations, since it takes care to avoid interpolating across boundaries. We take advantage of this ability by treating the borders of the orientation array as occluding boundaries, so that we may deal with objects which extend out of the image. For example, spherical surface orientations were correctly recovered from the partially visible boundary shown in Figure 10. The case of the tilted cylinder discussed above is a second example.

Experiments with occluded boundaries raised the question of just how little boundary information suffices to effect recovery. We experimented with a limiting case in which we attempted to reconstruct surface orientation of a sphere from just four initial boundary values at the corners of the arrays. This corresponds to the image of a large sphere whose boundary circumscribes the square array (see Figure 11). The resulting surface orientations produced from these extremely sparse initial conditions were as accurate as when all the boundary orientations are given, but more iterations were required. For example, fixing the Nx and Ny orientations at the corners of a 17 x 17 square array to the values for a sphere of radius 12, the maximum absolute error of the reconstructed interior orientations after 400 iterations was less than .005.

D. Qualitative Boundary Conditions

In all the above experiments, boundary conditions were provided by specifying exact orientations at all unoccluded points along extremal boundaries. The values of Nx and Ny at these points were initially inserted in the arrays and were held fixed through all iterations. In a complete visual system it is necessary to derive these values from the shape of extremal boundaries in the image. In principle, this can be done easily, since the surface normal at each point is constrained to be orthogonal to both the tangent to the boundary and to the line of sight. (For orthogonal projection, the normal must thus be parallel to the image plane.) In a spatially quantized image, the accurate determination of tangent is difficult, particularly when the object is not very large compared to the quantization grid.

One way to overcome this problem is to introduce the notion of qualitative, partially-constraining boundary conditions. We can, for example, constrain the surface normals along a quantized extremal boundary to be approximately parallel to the image plane and point outward across the boundary. We then rely on the iterative process to reconstruct exact values for the normals at points on the boundary, treating them just like interior points. To implement this approach, we introduce a step which at each iteration checks the orientation at boundary points. For each boundary element adjacent to the point, we check that the surface normal has a component directed outward across it. If it does not, the value of Nx or Ny is modified appropriately. The value of Nz is also checked to be close to zero, and vector N is normalized to ensure it remains a unit vector. This process was applied to the spherical, cylindrical, and elliptical test cases, and was found to yield orientation values accurate to 10 percent, for both interior and boundary points, after only 100 iterations. The principal limitation on accuracy appears to be the coarse quantization grid being used.

IX DISCUSSION

The ability to handle sparse or partially constrained initial conditions is important in a reconstruction algorithm because often nothing else is obtainable. Line drawing interpretation is an obvious example since surface orientation is constrained only along boundaries. In grey-level imagery, photometric constraints yield families of normals at most points on a smooth surface, not unique values. Even direct range measurement, as provided by stereo, motion parallax, and laser range-finders, may result in data that are noisy or missing in places.

Experimentation is continuing to determine how well the reconstruction technique performs, both in absolute terms and relative to human perception, for a variety of test surfaces. Of particular interest is whether the assumption of locally uniform curvature is an adequate basis for reconstruction. Simultaneously, we are investigating alternative interpolation operators that reflect measures of curvature appropriate to different surface types, such as soap films.

We are also extending the program to deal with a wider range of reconstruction problems, including, specifically, reconstruction from noisy range values and from partially constrained normals along intersection edges (which are constrained only to be orthogonal to a three-dimensional line element). These extensions will require properly integrating surface orientation and range (which may require making the integrability condition of Equation 5 explicit), and smoothing noisy, and possibly inconsistent, data. Ultimately, a general vision system will need the ability to add and delete hypothesized discontinuities so that surfaces and boundaries can be simultaneously refined.

Although the reconstruction process we have described is conceptually parallel, there are inherent limitations on how fast information can propagate across the image. Thus, convergence speed is of practical concern. Using larger operators increases the effective velocity of propagation but can impair precision where small features are involved. What seems to be required is a scheme that combines multiple sizes of operators in a hierarchical organization, where initial estimates provided by the larger operators are refined by the smaller ones. We are studying a number of theoretical questions raised by a hierarchical approach to surface reconstruction, including the effects of operator size on speed and accuracy, and the key question of how information propagates between levels of the hierarchy.

X CONCLUSION

Interpolating smooth surfaces from boundary conditions is a ubiquitous problem in early visual processing [1, 2, 7, 11-18]. We have described a solution for an important special case: the interpolation of surfaces that are locally spherical or cylindrical, given initial orientation values and constraints on orientation. We developed parallel computational techniques for reconstruction of such surfaces, exploiting the observation that, since curvature is uniform, the components of the unit normal vary linearly across the image.

Reconstruction experiments on spherical and cylindrical test cases have produced essentially exact reconstructions, even when boundary values were extremely sparse or only partially constrained. Results on other test cases seem in reasonable agreement with human perception.

REFERENCES

1. H. G. Barrow and J. M. Tenenbaum, "Recovering Intrinsic Scene Characteristics from Images," in Computer Vision Systems, A. Hanson and E. Riseman, eds., pp. 3-26 (Academic Press, New York, New York, 1978).

2. D. Marr, "Representing Visual Information," in Computer Vision Systems, A. Hanson and E. Riseman, eds. (Academic Press, New York, New York, 1978).

3. L. S. Davis and A. Rosenfeld, "Noise Cleaning by Iterated Local Averaging," IEEE Trans. SMC, Vol. 8, pp. 705-710 (1978).

4. R. Haralick, "A Facet Model for Image Data," Proc. IEEE Conference on Pattern Recognition and Image Processing, Chicago, Illinois, pp. 485-497 (August 1979).

5. L. Brand, Vector and Tensor Analysis (John Wiley, 1953).

6. H. G. Barrow and J. M. Tenenbaum, op. cit., p. 19, para. 4.

7. A. Witkin, "The Minimum Curvature Assumption and Perceived Surface Orientation," presentation at Optical Society of America, November 1978.

8. F. J. Almgren, Jr., and J. E. Taylor, "The Geometry of Soap Films and Soap Bubbles," Scientific American, pp. 82-93 (July 1976).

9. D. A. Huffman, "Curvature and Creases: A Primer on Paper," IEEE-TC, Vol. C-25, No. 10, (October 1976).

10. C. Wilcox, M. Dageforde, and G. Jirak, "MAINSAIL Language Manual," Stanford University, Stanford, California (July 1979).

11. D. Marr, "Analysis of Occluding Contour," Proc. Roy. Soc. Lond. B, Vol. 197, pp. 441-475 (1977).

12. K. Stevens, "Surface Perception From Local Analysis of Texture and Contour," Ph.D. Dissertation, Electrical Engineering and Computer Science, Mass. Inst. of Technology, Cambridge, Massachusetts.

13. H. G. Barrow and J. M. Tenenbaum, "Recovery of Three-Dimensional Shape Information from Image Boundaries," (in preparation).

14. B.K.P. Horn, "Obtaining Shape from Shading Information," in The Psychology of Computer Vision, P. H. Winston, ed. (McGraw-Hill, New York, New York, 1975).

15. R. Woodham, "A Cooperative Algorithm for Determining Surface Orientation from a Single View," Proc. Fifth Intl. Joint Conference on Artificial Intelligence, Cambridge, Massachusetts, pp. 635-641 (August 1977).

16. M. Brooks, "Surface-Normals from Closed Paths," Proc. Sixth Intl. Joint Conference on Artificial Intelligence, Tokyo, Japan, pp. 98-101 (August 1979).

17. D. Marr and T. Poggio, "Cooperative Computation of Stereo Disparity," Science, Vol. 194, pp. 283-287 (1977).

18. W. F. Clocksin, "Determining the Orientation of Surfaces from Optical Flow," Proc. AISB Conference on Artificial Intelligence, Hamburg, West Germany, pp. 93-102 (July 1978).

TABLE I - INTERPOLATION RESULTS FOR SPHERICAL TEST CASE

Operator	# Iterations	Max. Abs. Error	Average (RMS) error
		(Nx, Ny)	(Nx, Ny)
Uniformly Weighted Average over 4-connected 3x3 patch	50	.0165	.0075
	100	.0004	.0002
Uniformly Weighted Average over 8-connected 3x3 patch	50	.0007	.0003
	100	.0000006	.0000003
∇^2 over a 4-connected 3x3 patch	50	.006	.003
	100	.00006	.00003
8-way linear interpolation/extrapolation (see Figure 6)	50	.004	.002
	100	.00002	.00001
4-way linear interpolation/extrapolation (just parallel to x and y axes)	50	.03	.01
	100	.001	.0007
Gradient smoothing over a 4-connected 3x3 patch	50	.40	.19
	100	.26	.12
	200	.10	.05
Gradient smoothing over an 8-connected 3x3 patch	50	.13	.05
	100	.03	.01
	200	.001	.0005

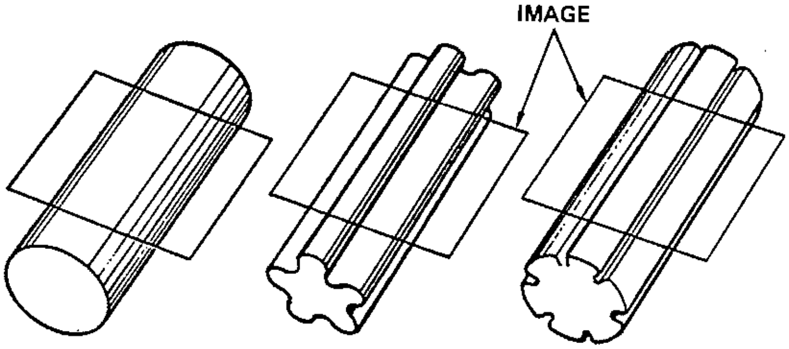

FIGURE 1

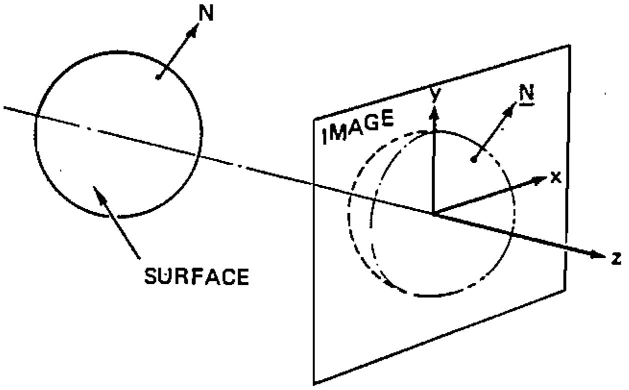

FIGURE 2

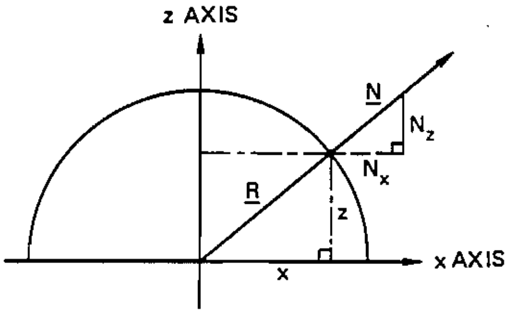

FIGURE 3

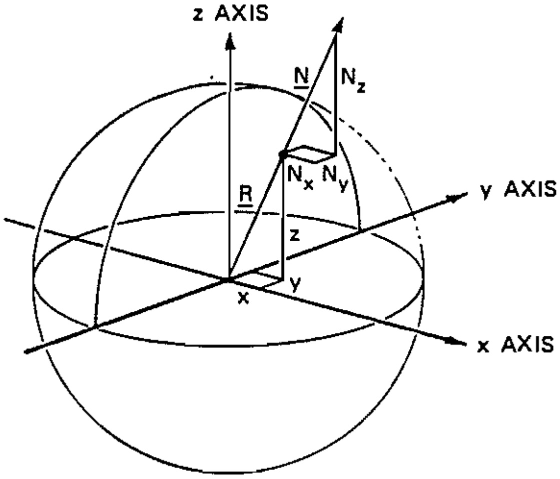

FIGURE 4

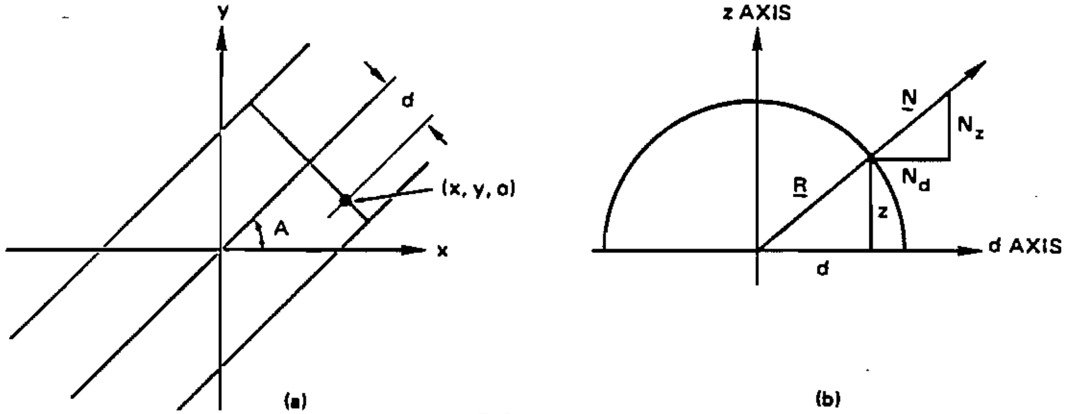

FIGURE 5

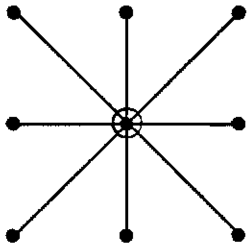

FIGURE 6

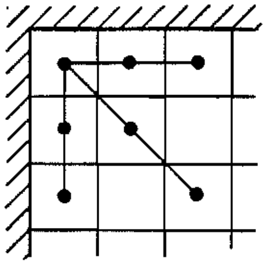

FIGURE 7

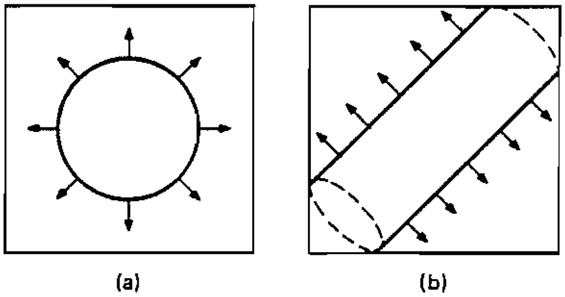

FIGURE 8

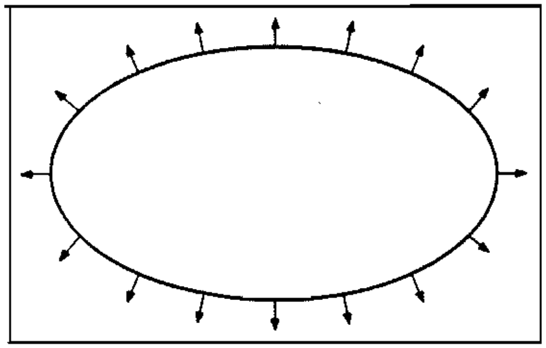

FIGURE 9

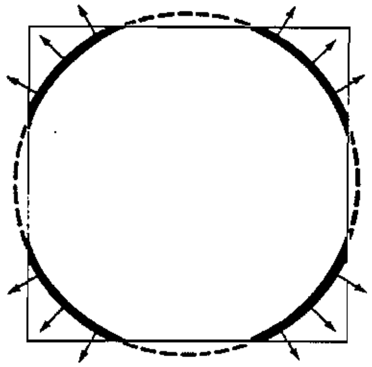

FIGURE 10

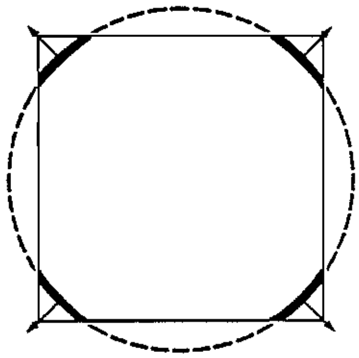

FIGURE 11